

<b>REPORT DOCUMENTATION PAGE</b>				<i>Form Approved</i> <b>OMB No. 0704-0188</b>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
<b>1. REPORT DATE (DD-MM-YYYY)</b> 10-12-2010		<b>2. REPORT TYPE</b> Article		<b>3. DATES COVERED (From - To)</b> DEC 2010 - JAN 2011	
<b>4. TITLE AND SUBTITLE</b> Rate Control for Network-Coded Multipath Relaying with Time-Varying Connectivity				<b>5a. CONTRACT NUMBER</b> FA8720-05-C-0002	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Brooke Shrader, Armen Babikyan, Nathaniel M. Jones, Thomas H. Shake, and Andrew P. Worthen				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> DDRE, 1777 N. Kent St. Rosslyn, VA 22209				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> OSD-ATL	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> This paper presents techniques for achieving high throughput in delay-constrained, multihop wireless communication networks with time-varying link connectivity. We develop a rate-controlled, multipath strategy using network coding, and compare its performance with that of multipath flooding and with the performance of traditional single-path strategies. These performance comparisons include both theoretical benchmarks and simulation results from cooperative relay scenarios, which incorporate different sets of link connectivity statistics that are drawn from field tests of mobile satellite communication terminals. The results indicate that with appropriate rate-control, network coding can provide throughput performance comparable to multipath flooding of the network while utilizing bandwidth nearly as efficiently as single-path routing.					
<b>15. SUBJECT TERMS</b> multipath routing, network coding, rate control, congestion control, delay					
<b>16. SECURITY CLASSIFICATION OF:</b> U			<b>17. LIMITATION OF ABSTRACT</b> SAR	<b>18. NUMBER OF PAGES</b> 11	<b>19a. NAME OF RESPONSIBLE PERSON</b> Zach Sweet
<b>a. REPORT</b> U	<b>b. ABSTRACT</b> U	<b>c. THIS PAGE</b> U			<b>19b. TELEPHONE NUMBER (include area code)</b> 781-981-5997

# Rate Control for Network-Coded Multipath Relaying with Time-Varying Connectivity

Brooke Shrader, Armen Babikyan, Nathaniel M. Jones, Thomas H. Shake, and Andrew P. Worthen

**Abstract**—This paper presents techniques for achieving high throughput in delay-constrained, multihop wireless communication networks with time-varying link connectivity. We develop a rate-controlled, multipath strategy using network coding, and compare its performance with that of multipath flooding and with the performance of traditional single-path strategies. These performance comparisons include both theoretical benchmarks and simulation results from cooperative relay scenarios, which incorporate different sets of link connectivity statistics that are drawn from field tests of mobile satellite communication terminals. The results indicate that with appropriate rate-control, network coding can provide throughput performance comparable to multipath flooding of the network while utilizing bandwidth nearly as efficiently as single-path routing.

**Index Terms**—multipath routing, network coding, rate control, congestion control, delay

## I. INTRODUCTION

Changes in link connectivity inherent to mobile wireless networks create challenges in carrying out high throughput, delay constrained communication, particularly because the communication rate must be adapted in time to fully utilize bandwidth while also limiting congestion. In this work we consider one instance of a wireless network with time-varying link connectivities and explore the use of path diversity coupled with network coding to achieve high throughput under delay constraints.

The practical problem setting that motivates our work is the transmission of data via satellite to a mobile terrestrial user. In this problem, data must be sent from a fixed terrestrial terminal via satellite to a mobile user  $d$  in a distant location. Transmissions to and from the satellite incur a large propagation delay (e.g., 125 msec) due to the distance packets must travel. Suppose further that node  $d$  is in the proximity of  $N$  other mobile users; these  $N+1$  nodes represent vehicles driving through an urban area where buildings can obstruct their connection to the satellite. Field tests of this scenario [11] show that the link between a mobile terrestrial vehicle and the satellite can be modeled by a two-state Markov model in which the mobile node is either blocked or unblocked from the satellite. Furthermore, the same field tests indicate that if there is sufficient spatial separation between a pair of mobile terrestrial nodes, their blockage processes are uncorrelated [11]; so when  $d$  is blocked from the satellite, one or more of

the  $N$  relay nodes may be unblocked and can transfer data to  $d$ . Field test data of link blockage for vehicles driving through a number of different environments – including urban, suburban, and rural settings – show mean link blockage durations ranging from one second to over ten seconds.

In the setting described above, the blockage duration is too long for techniques like forward error correction (FEC) and automatic repeat request (ARQ) to ensure reliability while still meeting a delay constraint of a few seconds. On the other hand, adaptive single-path routing protocols typically repair routes after tens of seconds, which is not fast enough to handle blockages of these durations. This motivates the use of multipath strategies, where multiple relays concurrently receive and relay data for a blocked destination node. One multipath strategy is for every relay node to collect and re-transmit all packets it receives from upstream; we refer to this strategy as multipath flooding and note that it can lead to inefficient use of bandwidth as duplicate copies of packets are carried over multiple paths. We focus instead on the use of random linear network coding, in which relays collect packets received from the satellite and transmit random linear combinations of those packets on terrestrial relay links.

Random linear network coding, introduced in [5], can support multipath relaying and provide robustness to dynamically varying connections. We assume that random linear network coding is carried out as outlined in [2]. Data packets arriving at a source node are grouped into generations or blocks, where each generation contains  $K$  packets. Nodes in the network can form and transmit coded packets, which are random linear combinations of packets from the same generation. The coefficients of each random linear combination are chosen randomly and uniformly from a large finite field and the coefficients mapping the coded packet to the original data packets are included in the header. The advantage of this strategy is that relay nodes on disjoint paths do not have to coordinate which packets to transmit and yet they can relay linearly independent data with high probability. The destination will be able to decode the generation if  $K$  or more coded packets arrive, provided that  $K$  of the received coded packets are linearly independent. The probability that any  $K$  coded packets received by the destination are linearly independent can be made large if the field size is chosen sufficiently large and if each relay collects enough packets from each generation before encoding and transmitting any packets downstream.

Previous work on network coding addresses the problems of subgraph selection, in which a subset of nodes in the network are selected to encode and forward packets, and rate

The authors are with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington MA.

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

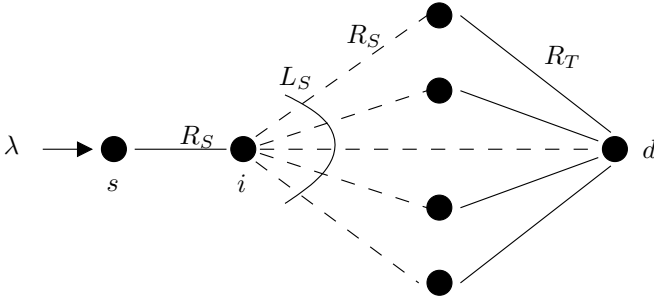


Fig. 1. Network setting under consideration. Data is to be sent from source  $s$  to destination  $d$  via intermediate node  $i$  and  $N$  relay nodes. In this figure  $N = 4$ .

control on the subgraph. Much of the previous work in this area, including [2], [9], focuses on the use of network coding for multicast traffic. Network coding can also be particularly efficient for unicast traffic; as shown in [8], it can outperform link-by-link and end-to-end ARQ and FEC strategies in terms of the number of transmissions in the network for every packet received at the destination. Obtaining this efficiency requires careful design of the rate at which coded packets are transmitted on every path. If coded packets are injected at a low rate on a path (e.g., too few packets sent per generation) then bandwidth may be under-utilized, path diversity may not be fully exploited, and the destination may not receive enough coded packets to decode; on the other hand, injecting coded packets at a high rate on a path may congest the network.

The works in [8], [12] tackle the rate control problem by developing algorithms that solve a constrained optimization problem to determine the rate at which innovative coded packets are transmitted on each link. This approach is effective in determining an optimal rate allocation; however, the link connectivities are assumed to be static, e.g., in [12] link qualities may change only on a daily basis. In our motivating problem setting, when link connectivities change, these rate-control algorithms would need to be re-run and the propagation delay will inhibit quick rate adaptation. A subgraph selection and rate control strategy for time-varying networks is developed in [9], but the authors assume that the source transmission rate has been set to a value that does not cause congestion. By contrast, in our work, we drop this assumption and handle the case in which the source rate can lead to congestion downstream. Finally, the rate control mechanism developed in [1] and used in [7] is also able to support a time-varying topology: periodic ping messages are used to determine link connectivities and loss probabilities, which are communicated to other nodes in the network; each node then computes the number of coded packets to send using its distance (in hops) to the destination and the loss probabilities on downstream links. This strategy can avoid congestion by scaling back the transmission rate all the way to the source. However, this approach may not be practical in our setting: the instantaneous capacity of the network may change nearly as frequently as the time needed to propagate data to and from a satellite.

In contrast to previous work, we develop rate-control strategies that work in a localized way at shorter timescales and avoid congestion by coordinating the dropping of generations

at intermediate relay nodes. By dropping coded generations at intermediate nodes, we avoid the need to scale back the transmission rate at the source, which would incur a large propagation delay. This congestion avoidance approach is inspired by packet dropping strategies used for congestion control on the internet [4] and is a novel contribution of our work. Through the congestion avoidance strategy, we trade-off the reception of some packets in order to ensure that packets received at the destination meet a delay constraint.

## II. PROBLEM SETTING

We consider the scenario shown in Figure 1. Unicast traffic generated by a (deterministic) periodic arrival process of rate  $\lambda$  bps needs to be sent from the source node  $s$  to destination node  $d$ . All data is sent through the network in the form of packets. Data is first transmitted from the source to an intermediate node  $i$  over a lossless link operating at constant rate  $R_S$  bps. Transmissions to and from node  $i$  incur a large propagation delay. The offered traffic from the source node may exceed  $R_S$ , and we let  $L_S$  denote the rate at which data arrives at the intermediate node, where  $L_S = \min(\lambda, R_S)$ . The data are then broadcast at rate  $L_S$  bps from the intermediate node to a group of  $N + 1$  nodes. The  $N + 1$  nodes that can receive the broadcast include  $N$  relay nodes and the destination node  $d$ . The links between the intermediate node  $i$  and each of the  $N + 1$  nodes are assumed to be independent and to incur identically distributed blockages. Specifically, we assume that each of these links follows the two-state Markov model shown in Figure 2. The link is either in the CLEAR or “good” state, for which all data transmitted by  $i$  is received without loss; or in the BLOCKED or “bad” state, for which all data transmitted by  $i$  is lost or erased. In the discussion that follows, we use the terms clear and unblocked interchangeably. The link connectivity model is parameterized by two values:  $p_{gg}$  denotes the probability of self-transition in the clear state and  $p_{bb}$  denotes the self-transition probability in the blocked state; we assume that the values  $p_{gg}$  and  $p_{bb}$  are the same for all  $N + 1$  links.

When the link from  $i$  to  $d$  is blocked, data can still be recovered through transmissions by the  $N$  relay nodes. We assume that each of the  $N$  relay nodes has a direct link to  $d$  that operates without loss at rate  $R_T$  bps. We also assume that there is no interference between links from relays to  $d$ ; this can be achieved by collision-free medium access control. Let  $N(t) \in [0, 1, \dots, N]$  denote the number of unblocked relay nodes (ie, number of relay nodes in the clear state) at time  $t$ ; clearly  $N(t)$  can be modeled as a stationary and ergodic Markov process. We assume that each relay node maintains a finite buffer for collecting packets sent by  $i$  and intended for  $d$ . Although the link between a relay node and  $d$  does not incur losses, we allow the possibility that some data can be lost or dropped at a relay node due to buffer overflow. The buffer overflow and associated packet dropping problem is particularly relevant when  $R_T < R_S$ ; this is the case we consider in much of our work.

Our objective in this setting is to maximize the throughput from  $s$  to  $d$  while meeting a delay constraint; we would like to

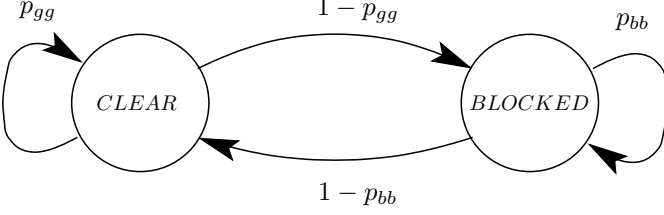


Fig. 2. Two-state Markov model for link connectivity. In the BLOCKED state all transmitted data are lost or erased; in the CLEAR state all transmitted data are received without error.

achieve this objective over a wide range of values of offered load  $\lambda$ . Note that the heterogeneity in link rates  $R_S$  and  $R_T$  can lead to congestion, which is problematic for meeting a delay constraint. For instance, when  $R_T < R_S$  and node  $d$ 's link is blocked, the relays can deliver data to  $d$  at a rate of at most  $R_T N(t)$ ; if  $L_S > R_T N(t)$  then timely delivery of all data generated at  $s$  may not be possible. In general our objective is at all times to efficiently use the *instantaneous capacity* of the network. Let  $I(t)$  be an indicator function that takes value 1 if the link between  $i$  and  $d$  is clear at time  $t$ . If  $I(t) = 1$  then the network can support at most  $R_S$  bps from  $s$  to  $d$ . If  $I(t) = 0$  then the network can support at most  $\min(R_S, R_T N(t))$  bps from  $s$  to  $d$ . The instantaneous capacity from  $s$  to  $d$  at time  $t$  is denoted  $C(t)$  and given as follows.

$$C(t) = \max(R_S I(t), (1 - I(t)) \min(R_S, R_T N(t))) \quad (1)$$

In our motivating problem setting,  $C(t)$  may change on the order of every second. Our aim is to achieve a throughput given by the minimum of the offered load  $\lambda$  and the instantaneous capacity  $C(t)$ .

### III. PERFORMANCE PREDICTION

In this section we provide benchmarks for the average throughput achievable by three different strategies. As described above, the link states and the number of unblocked relays follow a random process with memory. In order to develop simple performance benchmarks, we will consider the long-term average behavior of links. Let  $p_g$  denote the stationary probability that a link is in the clear state; similarly  $p_b$  denotes the stationary probability of a link in the blocked state. These are given by

$$p_g = \frac{1 - p_{bb}}{2 - p_{bb} - p_{gg}}, \quad p_b = 1 - p_g. \quad (2)$$

The number of relays with a clear link  $N(t)$  also has a stationary distribution; since  $N(t)$  is the sum of  $N$  independent link states, each of which is clear with probability  $p_g$ , its stationary distribution is given by the binomial distribution as follows.

$$\pi_i = \lim_{t \rightarrow \infty} \Pr(N(t)=i) = \binom{N}{i} p_g^i p_b^{N-i}, \quad i = 0, 1, \dots, N \quad (3)$$

Below we develop approximations for the throughput in bps for single-path routing, multipath flooding, and multipath

network coding. These approximations do not account for a specific delay constraint. However, an underlying assumption is that intermediate and relay nodes cannot store up large buffers of packets to be transmitted over a long period of time, and as such the delay is given by the propagation time plus the time it takes to empty a small queueing buffer. In the following we describe approximations for the throughput in bps denoted  $S$ . We also benchmark the fraction of packets arriving at  $s$  that are received and decoded at  $d$ ; we refer to this quantity as completion rate and compute it as  $S/\lambda$ .

#### A. Single-path routing

We can upper bound the throughput achievable by a single-path routing strategy by assuming that a genie can notify all nodes in the network of the highest-rate path between  $s$  and  $d$  at any point in time and that this can be done without error and without delay. If  $L_S \leq R_T$ , then it is possible to deliver all  $L_S$  bps to  $d$  without buffer overflows at any relay node, provided that at least one of the  $N + 1$  nodes has a clear link, which happens with probability  $1 - p_b^{N+1}$ . However, if  $L_S > R_T$ , then a relay node will only be able to deliver a fraction of the traffic. In this case, if  $d$  has a clear link, then the entire  $L_S$  bps can be delivered; if  $d$  has a blocked link but at least one of the  $N$  relay nodes has a clear link, then data can be delivered to  $d$  at rate of  $R_T$  bps. Our upper bound on the throughput of a single-path routing protocol is denoted  $S^{SPR}$  and is given as follows.

$$S^{SPR} = \begin{cases} L_S (1 - p_b^{N+1}), & L_S \leq R_T \\ L_S p_g + R_T p_b (1 - p_b^N), & L_S > R_T. \end{cases} \quad (4)$$

#### B. Multipath flooding

Next we develop an approximation for the throughput when a flooding technique is used by the relays. Specifically, we assume that each relay node collects all data it is able to receive from  $i$  and aims to immediately replicate and forward all of that data to  $d$ . For  $L_S \leq R_T$ , the destination will be able to receive the  $L_S$  bps broadcast by  $i$  for a fraction of time corresponding to the probability that at least one of the  $N + 1$  nodes receives the data, which happens with probability  $1 - p_b^{N+1}$ . For  $L_S > R_T$ , we have two cases. First, if  $d$  is able to receive directly from  $i$ , which happens with probability  $p_g$ , it will do so at rate  $L_S$ . If  $d$  has a blocked link, which happens with probability  $p_b$ , then data will be relayed by at most  $N$  nodes, but some packets will be dropped due to buffer overflows that occur at each relay. We make the assumption that only a fraction  $R_T/L_S$  of the data can be sent to  $d$  by each relay, and relays independently and uniformly drop packets arriving from  $i$ . Each of the  $N$  nodes will be able to relay data with probability  $1 - p_b$  and on average, a portion  $1 - (1 - p_b)R_T/L_S$  of the  $L_S$  bps sent by  $i$  will be dropped on each of the  $N$  paths. The throughput for this flooding strategy is approximated by

$$S^{FL} = \begin{cases} L_S (1 - p_b^{N+1}), & L_S \leq R_T \\ L_S \left( p_g + p_b \left( 1 - \left( 1 - \frac{(1 - p_b)R_T}{L_S} \right)^N \right) \right), & L_S > R_T. \end{cases} \quad (5)$$

### C. Multipath network coding

We develop an upper bound on the performance of network coding by assuming that every coded packet sent by a relay node is linearly independent of all packets sent by other relay nodes; in other words, each path from a relay node to  $d$  can carry unique information at a rate given by the link rate  $R_T$ . This would be equivalent to achieving the max-flow min-cut capacity, which is shown to be accomplished by network coding in [3], where cut-capacity is defined to incorporate broadcast wireless transmission such as that performed by node  $i$ . Achieving this would require performing network coding over an infinitely-large finite field (which would ensure that no relay ever generates a linearly dependent packet) as well as ensuring that no more than  $K$  packets per generation are received by  $d$  from the collection of relay nodes. Moreover, we assume that the dwell time in the clear state of a link is much larger than the time it takes for a generation of  $K$  packets to be broadcast over the link emanating from node  $i$ ; the time it takes for a generation to be sent by  $i$  depends on  $K$ ,  $R_S$ , and  $\lambda$ , so if  $K$  is too large,  $R_S$  is too small, or  $\lambda$  is too small, this assumption breaks down. Under this assumption, we take  $p_g$  to represent the steady-state probability that an *entire generation* of packets is received from  $i$ .

When  $d$  has a blocked link, the relay nodes can provide data to  $d$  at a rate of at most  $N(t)R_T$ . We can use the steady-state distribution of  $N(t)$  to obtain an upper bound on the throughput given by

$$L_S p_g + p_b \sum_{i=1}^N \pi_i \min(L_S, iR_T) \quad (6)$$

For  $L_S \leq R_T$ , the expression above is equivalent to the throughput of flooding in Eqn. (5); for  $L_S > R_T$ , it can be shown by induction on  $N$  that the expression above is greater than or equal to the flooding throughput  $S^{FL}$  in (5). In addition, random linear network coding may require the use of packet headers to identify generations and encoding vectors for each packet; we augment the expression above to account for this packet overhead. Let  $B_{pkt}$  denote the size in bytes of packets to be delivered to the destination (this is the same as the packet size used in flooding) and  $B_{hdr}$  denote the size in bytes of packet headers used for random linear network coding. To account for network coding overhead, we compute throughput by scaling back the bps rates that traffic can be carried on links by substituting the values

$$R_S^{NC} = R_S \frac{B_{pkt}}{B_{pkt} + B_{hdr}}, \quad R_T^{NC} = R_T \frac{B_{pkt}}{B_{pkt} + B_{hdr}}. \quad (7)$$

Finally, the predicted throughput for multipath network coding is given by

$$S^{NC} = L_S^{NC} p_g + p_b \sum_{i=1}^N \pi_i \min(L_S^{NC}, iR_T^{NC}) \quad (8)$$

where  $L_S^{NC} = \min(\lambda, R_S^{NC})$ . To compare the throughput of multipath network coding with multipath flooding, we should compare the expression for  $S^{FL}$  in Eqn. (5) with the expression for  $S^{NC}$  given in (8). For  $L_S \leq R_T$  and  $B_{hdr} > 0$ , the throughput  $S^{FL}$  of flooding is strictly greater than the

throughput  $S^{NC}$  of multipath network coding. However, for  $L_S > R_T$ , the comparison between the two depends largely on the size of the packet header relative to the size of the packet. When  $L_S > R_T$ , the network coding throughput  $S^{NC}$  can be larger than the flooding throughput  $S^{FL}$  provided  $B_{hdr} \ll B_{pkt}$ ; this is the case considered in our experimental scenario and numerical results are provided in Section V.

### IV. RATE-CONTROL FOR NETWORK CODING

Our approach to rate-control is as follows. First, the destination  $d$  forms an estimate of the available capacity  $C(t)$  for the flow. If  $d$  is blocked, the  $N(t)$  relay nodes are assigned to transmit data to  $d$  at equal rates. If the offered traffic  $L_S$  sent by node  $i$  exceeds the available relay capacity  $N(t)R_T$ , the congestion avoidance mechanism works to identify generation IDs for packets to be discarded at the relays. Through congestion avoidance, our approach adapts the rate locally, rather than adapting the transmission rate all the way back at the source and incurring a large propagation delay. Additionally, through congestion avoidance, we ensure that the flow uses only the currently-available capacity in the network, so that rather than waiting in long queues at relay nodes, packets delivered to the destination meet a delay constraint.

#### A. Basic operation

We assume that network coding is performed by allowing multiple generations to propagate through the network at the same time without requiring any end-to-end acknowledgments for decoded generations. Each relay node maintains a value for  $I(t)$  that specifies whether any data relay to  $d$  is necessary. When  $I(t) = 1$  no relaying is necessary and the relays simply monitor control messages. When  $I(t) = 0$ , relaying of information is necessary and relays enable the following logical flow. Each relay that has a clear link collects packets from node  $i$  until one or more full generations are collected. We require the collection of full generations before any coded packets are sent in order to increase the probability that linearly independent coded packets are sent on each path. For each full generation of packets collected, the relay first decides whether or not to forward any packets from that generation. If packets are to be forwarded, the relay then decides how many packets from the generation to encode and enqueue for transmission.

There is a separate flow for control information. Node  $d$  maintains information on the local state of the network, which it transmits to the relays when the state changes. The destination maintains values for the number of connected relays  $N(t)$ , the rate  $L_S$  that data is sent from node  $i$ , and  $I(t)$  indicating the connectivity of the direct link from  $i$  to  $d$ . The destination monitors its own link to determine  $I(t)$  and may receive control packets from the relays with estimates of  $N(t)$  and  $L_S$ . Whenever any of these values changes, the destination sends control messages to the relays updating the values. The discussion below specifies the operations of relay nodes assuming that  $I(t) = 0$ .

We split the rate among paths by specifying the number of coded packets each relay sends for each generation. Assuming

that there are no losses on links between a relay and  $d$  and assuming that relays share equally the burden of forwarding traffic, each relay should send  $K/N(t)$  packets per generation. When  $K/N(t)$  is an integer, the specified operation is clear. When  $K/N(t)$  is not an integer, each relay must either send more than  $K/N(t)$  packets, causing inefficiency, or must send fewer than  $K/N(t)$  packets, with some fraction of relays transmitting extra packets to make up the difference. In the protocols described below one of the following options is performed.

- **Deterministic Rate Splitting:** Each relay enqueues

$$\left\lceil \frac{K}{N(t)} \right\rceil \text{ packets/generation} \quad (9)$$

- **Probabilistic Rate Splitting:** Each relay enqueues

$$\left\lceil \frac{K}{N(t)} \right\rceil + X \text{ packets/generation, } X \sim \text{Bernoulli}(p) \quad (10)$$

For Probabilistic Rate Splitting,  $p$  is a tunable parameter that is specified below. We note that Deterministic Rate Splitting is the strategy used in [9] for multicast traffic. Deterministic Rate Splitting will ensure that at least  $K$  packets arrive at  $d$  and the generation can be decoded; however, this may come at the cost of extra redundant packets that congest the network. Probabilistic Rate Splitting may avoid sending redundant packets and the associated congestion, but there is often a positive probability that the destination does not receive enough coded packets to decode.

We also implement a packet discard algorithm for congestion avoidance. An important observation is that *because incomplete generations cannot be decoded by the destination, an effective packet discard strategy must coordinate the discard of entire generations*. Let  $C_R(t) = N(t)R_T$  denote the instantaneous capacity available for relays to forward data to  $d$ . When  $L_S > C_R(t)$  some generations will be discarded. The mechanism by which this is carried out is designed jointly with the rate splitting strategy.

### B. Congestion avoidance

When the offered traffic  $L_S$  is large relative to the relay capacity  $C_R(t)$ , we must ensure that each relay drops the same generations. We employ a technique inspired by a *numerically controlled oscillator* (NCO). The algorithm operates as follows. Using knowledge of  $K$ ,  $L_S$ ,  $R_T$  and  $N(t)$ , the overall fraction of incoming generations that can be successfully forwarded to the destination can be determined. This fraction will be approximated by a ratio of integers. The denominator is denoted  $baseNS$ , which is essentially a “granularity parameter”, and is chosen to be large enough for a close approximation, but small enough to allow an efficient implementation. The numerator is denoted  $stepNS$ , and is chosen to give the closest approximation to the desired fraction of generations to be forwarded given the fixed choice of  $baseNS$ . Each packet and generation has a generation identity number  $genID$ , a sequence number assumed here to be generated at the source. When  $C_R(t)$  is not sufficient to forward all received generations to the destination, an NCO

equation is used to decide which generations to discard and which to transmit. If

$$((genID \times stepNS) \bmod baseNS) < stepNS \quad (11)$$

then the generation is relayed; otherwise it is discarded. Since the generation IDs are generated prior to receipt by the relays, this rule is self-synchronizing and robust against link outages as long as all relays use the same values of  $stepNS$  and  $baseNS$ .

When  $L_S \leq C_R(t)$ , it may be possible to successfully transmit all received generations to the destination, in which case no decisions need to be made about which generations to discard and which to encode. Relay capacity constraints allow successful transmission of all received generations to  $d$  when  $L_S \leq C_R(t)$  and  $K/N(t)$  is an integer. However, when  $K/N(t)$  is not an integer, the Deterministic Rate Splitting strategy will send more than  $K/N(t)$  packets per generation, and this uses extra transmission capacity by a factor of  $\lceil K/N(t) \rceil / (K/N(t))$ . It is convenient to define the following ratios.

$$LCRatio = \frac{L_S}{C_R(t)} \quad (12)$$

$$QLCRatio = LCRatio \frac{\lceil K/N(t) \rceil}{K/N(t)} \quad (13)$$

For Deterministic Rate Splitting,  $QLCRatio$  is the threshold that determines when relay capacity is sufficient to successfully forward all generations.

If Deterministic Rate Splitting is used, then each relay must know whether  $QLCRatio \leq 1$  or whether  $QLCRatio > 1$ . However, if Probabilistic Rate Splitting is used, it may be reasonable to encode every generation even when  $QLCRatio > 1$ , provided that  $LCRatio \leq 1$  still holds. When  $LCRatio > 1$ , neither Deterministic nor Probabilistic Rate Splitting can attain efficient transmission of all generations to the destination, and some generations must be discarded. To implement effective rate control over a large range of offered loads, we will blend the Deterministic and Probabilistic Rate Splitting approaches. For this reason we introduce the following capacity ranges.

- **Capacity Range 1:**  $QLCRatio \leq 1$ ;
- **Capacity Range 2:**  $LCRatio \leq 1 < QLCRatio$ ;
- **Capacity Range 3:**  $LCRatio > 1$ .

The behavior of the rate control strategy will vary depending on which capacity range the network is operating in.

### C. Design and average behavior

In each of the three capacity ranges listed above, the protocol specifies a value of  $stepNS$ , which determines the fraction of received generations that are forwarded, and the rate-splitting strategy. These two specifications determine the actions taken by each relay in forwarding generations to  $d$ . The value of  $stepNS$  and the rate-splitting strategy are chosen to fully utilize available capacity  $C_R(t)$  while avoiding buffer overflows at relay nodes.

First, we introduce two tunable redundancy parameters used in specifying the protocol operation. Let  $\alpha$  be a *per generation*

*redundancy factor* used with Probabilistic Rate Splitting to increase the probability that collectively the  $N(t)$  relay nodes transmit  $K$  packets per generation to  $d$ . The value of  $\alpha$  is chosen so that the destination receives on average  $\alpha K$  coded packets (with one exception as noted below); setting  $\alpha = 1$  corresponds to adding no redundancy. Also we define  $\beta$  to be a *rate overload factor* that scales up the rate at which each relay node enqueues packets, potentially above the rate  $R_T$  of its outgoing link. In general these redundancy parameters are chosen such that  $1 \leq \beta \leq \alpha \leq 2$ .<sup>1</sup>

Next we describe the rationale behind jointly choosing  $stepNS$  and the rate-splitting strategy. For a fixed value of  $N(t)$ , let  $R_e$  denote the average rate at which a connected relay node enqueues packets to be sent to  $d$ . If relay nodes share equally the burden of sending packets, we would like  $R_e$  to be equal to  $L_S/N(t)$ . However, to avoid buffer overflows at relay nodes,  $R_e$  should be no more than  $R_T$ . Accounting for the rate overload factor, this dictates choosing  $stepNS$  and the rate-splitting strategy such that

$$R_e = \beta \min(R_T, L_S/N(t)). \quad (14)$$

Next, let  $F_G$  be the average ratio of generations forwarded from a relay to generations received by the relay from node  $i$ , and let  $P_G$  be the average number of packets per generation that each connected relay encodes and enqueues for each of these forwarded generations. The choice of  $stepNS$  and rate-splitting strategy determine  $F_G$  and  $P_G$ . Specifically,  $F_G = stepNS/baseNS$ . For Deterministic Rate Splitting,  $P_G = \lceil K/N(t) \rceil$ , while for Probabilistic Rate Splitting  $P_G = \lfloor K/N(t) \rfloor + p$ . These quantities are related to the per-relay enqueueing rate as follows.

$$R_e = P_G F_G L_S / K \quad (15)$$

In the four protocol variations described below,  $stepNS$  and the rate-splitting strategy are chosen by setting Equation (14) equal to Equation (15), with exceptions as noted below. The four protocol variations are summarized in Table I.

#### D. Four protocol variations

1) **Deterministic Protocol** : In this case Deterministic Rate Splitting is used in all capacity ranges. We set  $\alpha = \beta = 1$  because with Deterministic Rate Splitting, at least  $K$  packets will always be enqueued for transmission to the destination. This protocol ensures that every generation that is forwarded by the relays can be decoded by the destination. Note that in Capacity Range 1, if  $\lceil K/N(t) \rceil$  is not an integer, then the average per-relay enqueueing rate may exceed  $\min(R_T, L_S/N(t))$ . In Capacity Ranges 2 and 3, when  $N(t)\lceil K/N(t) \rceil$  is significantly greater than  $K$  packets per generation, this protocol will use capacity inefficiently and may incur buffer overflows that reduce throughput.

- **Capacity Range 1**  $QLCRatio \leq 1$ 
  - Set  $stepNS = baseNS$  (forward every generation)

<sup>1</sup>In many cases for Probabilistic Rate Splitting,  $p$  is chosen as  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$ , in which case  $0 \leq p \leq 1$  implies  $1 \leq \alpha \leq \frac{(1 + \lfloor \frac{K}{N(t)} \rfloor)}{\frac{K}{N(t)}} \leq 2$ . Also  $\beta \leq \alpha$  ensures that  $stepNS \leq baseNS$ .

- Deterministic Rate Splitting

- **Capacity Ranges 2 and 3**  $QLCRatio > 1$

- Set  $stepNS = \lfloor \frac{baseNS}{QLCRatio} \rfloor$
- Deterministic Rate Splitting

2) **Probabilistic Protocol** : In this case Probabilistic Rate Splitting is used in all capacity ranges and the choice of  $\alpha$  determines a tradeoff between increasing the probability that a complete generation is received at  $d$  and limiting congestion.

- **Capacity Ranges 1 and 2**  $LCRatio \leq 1$

- Set  $stepNS = \lfloor \frac{\beta}{\alpha} baseNS \rfloor$
- Probabilistic Rate Splitting with  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$

- **Capacity Range 3**  $LCRatio > 1$

- Set  $stepNS = \lfloor \frac{\beta}{\alpha} \frac{baseNS}{LCRatio} \rfloor$
- Probabilistic Rate Splitting with  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$

3) **Hybrid1 Protocol** : In the Probabilistic Protocol, the number of packets enqueued by the set of connected relays can be chosen so that on average the destination receives  $K$  packets, but variations from the average will cause performance degradations. The degradation may be particularly severe because the destination usually loses all  $K$  packets in a generation if even one fewer than  $K$  coded packets are received for that generation. This degradation is avoided by Deterministic Rate Splitting, and there is enough capacity to support this strategy in Capacity Range 1. Deterministic Rate Splitting can be used in Capacity Range 1, while Probabilistic Rate Splitting is helpful for reducing congestion in Capacity Ranges 2 and 3. This is the strategy used in the Hybrid 1 Protocol.

- **Capacity Range 1**  $QLCRatio \leq 1$

- Set  $stepNS = baseNS$  (forward every generation)
- Deterministic Rate Splitting

- **Capacity Range 2**  $LCRatio \leq 1 < QLCRatio$

- Set  $stepNS = \lfloor \frac{\beta}{\alpha} baseNS \rfloor$
- Probabilistic Rate Splitting with  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$

- **Capacity Range 3**  $LCRatio > 1$

- Set  $stepNS = \lfloor \frac{\beta}{\alpha} \frac{baseNS}{LCRatio} \rfloor$
- Probabilistic Rate Splitting with  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$

4) **Hybrid2 Protocol** : The Hybrid2 Protocol is similar to Hybrid1, but can improve performance in Capacity Range 2. In this range, Hybrid1 dictates that relays transmit a total of  $\alpha K$  packets per forwarded generation on average, but this does not make use of all the transmission capacity available from the  $N(t)$  connected relays. The Hybrid2 protocol makes use of this observation by allowing each relay to transmit at its maximum transmit rate, on average. This adds extra redundancy which may help increase throughput. Thus the difference between the Hybrid1 Protocol and Hybrid2 is that, in Capacity Range 2, Hybrid2 calculates the value of  $p$  to be slightly greater than

TABLE I

SUMMARY OF FOUR PROTOCOL VARIATION CHARACTERISTICS:  $P_G$  DENOTES THE AVERAGE NUMBER OF PACKETS PER GENERATION ENQUEUED BY A CONNECTED RELAY,  $F_G$  DENOTES THE AVERAGE RATIO OF RECEIVED GENERATIONS TO FORWARDED GENERATIONS, AND  $R_e$  DENOTES THE AVERAGE RATE IN BPS AT WHICH EACH CONNECTED RELAY ENQUEUES PACKETS.

		$P_G$	$F_G$	$R_e$
Deterministic	Cap. Range 1	$\lceil K/N(t) \rceil$	1	$\lceil \frac{K}{N(t)} \rceil \frac{L_S}{K}$
	Cap. Range 2, 3	$\lceil K/N(t) \rceil$	$\frac{K/N(t)}{\lceil K/N(t) \rceil \times LCRatio}$	$R_T$
Probabilistic	Cap. Range 1, 2	$\alpha K/N(t)$	$\frac{\beta}{\alpha}$	$\beta L_S/N(t)$
	Cap. Range 3	$\alpha K/N(t)$	$\frac{\beta}{\alpha \times LCRatio}$	$\beta R_T$
Hybrid 1	Cap. Range 1	$\lceil K/N(t) \rceil$	1	$\lceil \frac{K}{N(t)} \rceil \frac{L_S}{K}$
	Cap. Range 2	$\alpha K/N(t)$	$\frac{\beta}{\alpha}$	$\beta L_S/N(t)$
	Cap. Range 3	$\alpha K/N(t)$	$\frac{\beta}{\alpha \times LCRatio}$	$\beta R_T$
Hybrid 2	Cap. Range 1	$\lceil K/N(t) \rceil$	1	$\lceil \frac{K}{N(t)} \rceil \frac{L_S}{K}$
	Cap. Range 2	$(\frac{\alpha}{LCRatio}) \left( \frac{K}{N(t)} \right)$	$\frac{\beta}{\alpha}$	$\beta R_T$
	Cap. Range 3	$\alpha K/N(t)$	$\frac{\beta}{\alpha \times LCRatio}$	$\beta R_T$

the value used in the Probabilistic and Hybrid1 Protocols.<sup>2</sup>

- **Capacity Range 1**  $QLCRatio \leq 1$ 
  - Set  $stepNS = baseNS$  (forward every generation)
  - Deterministic Rate Splitting
- **Capacity Range 2**  $LCRatio \leq 1 < QLCRatio$ 
  - Set  $stepNS = \left\lfloor \frac{\beta}{\alpha} baseNS \right\rfloor$
  - Probabilistic Rate Splitting with  $p = (\alpha/LCRatio)(K/N(t)) - \lfloor K/N(t) \rfloor$
- **Capacity Range 3**  $LCRatio > 1$ 
  - Set  $stepNS = \left\lfloor \frac{\beta}{\alpha} \frac{baseNS}{LCRatio} \right\rfloor$
  - Probabilistic Rate Splitting with  $p = \alpha(K/N(t)) - \lfloor K/N(t) \rfloor$

## V. EXPERIMENTAL RESULTS

### A. Scenario and implementation

Our experimental results address the problem described in the Introduction: data is to be sent from a fixed terrestrial terminal over a lossless uplink to a satellite that relays data to a distant location. The satellite broadcasts data for a single unicast flow intended to be received at a single mobile terminal  $d$  which represents a vehicle driving in an urban area. This is the scenario shown in Fig. 1 and we evaluate the scenario assuming  $R_S = 5$  Mbps,  $R_T = 750$  kbps,  $N = 6$  relays, and a ground-to-satellite propagation delay of 125 msec to or from the satellite (i.e., 250 msec including both uplink

and downlink propagation). Application data arrives at the source node through periodic arrivals of fixed-length packets; in most test scenarios application packets consist of 1400 bytes (one exception is noted below) and are sent over UDP, which prepends a 28-byte header. The source and satellite nodes have an output queue for transmission that allow for buffering up at most 100 packets; clearly when  $\lambda > R_S$  the buffer at the source overflows and packets are dropped. The  $N = 6$  relay nodes can buffer at most 50 packets in an output queue for transmission to the destination and packets must be received at the destination within a deadline of 4 seconds after their arrival at the source. We have also tested relay buffer sizes smaller than 50 packets and deadlines shorter than 4 seconds as described below.

Link connectivities follow the two-state Markov model developed in [11]. The two-state Markov chain shown in Fig. 2 evolves in discrete space at intervals of one meter. We assume that the destination and  $N$  relays move at a constant speed of 30 miles per hour; this determines the duration of the time spent in each state. We consider three different variations of the blockage channel.

- **Moderate blockage:**  $p_{gg} = 0.9919$ ,  $p_{bb} = 0.9866$ ,  $p_g = 0.62$ , average blocked duration 5.6sec, average unblocked duration 9.2sec
- **Mild blockage:**  $p_{gg} = 0.9898$ ,  $p_{bb} = 0.9479$ ,  $p_g = 0.84$ , average blocked duration 1.4sec, average unblocked duration 7.3sec
- **Severe blockage:**  $p_{gg} = 0.9502$ ,  $p_{bb} = 0.9941$ ,  $p_g = 0.11$ , average blocked duration 12.6sec, average unblocked duration 1.5sec

These three blockage models are tested under constant link

<sup>2</sup>This changes the bounding values for  $\alpha$  slightly in Capacity Range 2. The Hybrid2 bounds in this region are  $1 \leq \alpha \leq LCRatio \frac{(1 + \lfloor \frac{K}{N(t)} \rfloor)}{(\frac{K}{N(t)})}$ , or, bounding over all possible values of  $K$  and  $N(t)$ ,  $1 \leq \alpha \leq 2LCRatio$ .



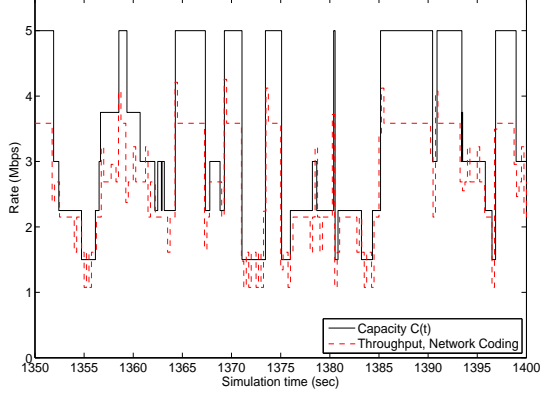


Fig. 3. Instantaneous Capacity  $C(t)$  and throughput achieved by the Deterministic Network Coding protocol versus simulation time for offered load  $\lambda = 3.6$  Mbps and Moderate Blockage.

rates  $R_S$  and  $R_T$  and constant number of relay nodes  $N$ ; the upper bounds on achievable throughput in these cases are very different.

Network coding has been implemented as an IP overlay in which node  $s$  in Fig. 1 is the overlay ingress node, node  $d$  is the overlay egress node, and all other nodes participate in the overlay throughout the entire simulation. All application packets are sent over UDP, which appends a 28-byte header; in the results shown below  $B_{pkt} = 1428$  bytes (unless noted otherwise). The network coding IP overlay appends an additional  $B_{hdr} = 52 + K$  bytes of header which is used to identify flows and generations. Every  $K$  packets arriving at node  $s$  are collected into one generation and marked with the same generation ID. Generations are sent sequentially through the network without any form of end-to-end acknowledgments; at any time there may be multiple generations propagating through the network. Every node in the network can store in memory packets for at most 100 generations at a time; if the memory is full and a packet for a new generation arrives, the oldest generation is flushed, or dropped. The  $N$  relay nodes follow the procedures specified by the protocols above; they can only transmit coded packets after a generation becomes complete (full rank). Both node  $s$  and node  $i$  perform systematic encoding [10] and send exactly  $K$  uncoded packets from a generation; this reduces delay and decoding complexity when the destination is unblocked. Network coding is performed using a field size  $GF(2^8)$  and generation size  $K = 8$  (except where noted otherwise).

To demonstrate the time-varying connectivity of the scenario and the adaptive features of our protocol, in Fig. 3 we display the instantaneous capacity  $C(t)$  as given in Eqn. (1) and the throughput achieved by the Deterministic rate control protocol for a 50 sec period of the simulation. This example is provided for the Moderate blockage case and as the figure shows, the capacity  $C(t)$  can change as often as once per second. Our protocol works to estimate available capacity and to adapt the rate in order to utilize available capacity. The throughput achieved by the Deterministic protocol is plotted in 0.25 sec bins, which causes the value shown on the plot to occasionally

spike above  $C(t)$ . We also note that when the link from  $i$  to  $d$  transitions from the blocked to clear state (i.e., when  $C(t)$  transitions from a value strictly less than  $R_S = 5$  Mbps to  $R_S$ ), there is a brief surge in throughput above the value of offered load  $\lambda = 3.6$  Mbps. This is due to the relay nodes emptying their buffers while  $d$  is concurrently receiving new packets from the satellite.

All results below show average performance over one-hour simulation runs. In the figures, the horizontal axis shows offered load  $\lambda$  in Mbps in terms of the arrival rate of application packets with UDP headers. Many of the results show packet completion rate on the vertical axis; this is computed by averaging the Mbps rate of throughput received within the deadline and then dividing by  $\lambda$ . We also display results on the protocol efficiency; this is computed by counting up the number of unique packets received (and decoded for network coding) at the destination and dividing by the total number of packets received (including partial generations for network coding). We display the predicted performance given by  $S^{SPR}$ ,  $S^{FL}$ , and  $S^{NC}$  as lines as well as the simulation results, which are shown as lines with symbols.

#### B. Comparison of four protocol variations

Figs. 4 and 5 show results comparing the four different rate control protocols presented in this paper; these figures display behaviors of network coding that we observed in many simulation runs. The lowest data point of offered load shown here corresponds to packets arriving once per second. At this arrival rate, none of the relay nodes are able to collect an entire generation of  $K \geq 8$  packets before exceeding the delay constraint; for this reason the packet completion rate reflects only what the destination  $d$  is able to receive when it is unblocked and is approximately equal to  $p_g$ . In general, the completion rate performance of network coding at low offered loads is poor because of the inability of relay nodes to collect a complete generation within the deadline and before transitioning from an unblocked to blocked state. As the offered load increases, our network coding implementation performs nearly as well as predicted by  $S^{NC}$ . As the offered load  $\lambda$  approaches  $R_S$ , the network becomes congested and the buffers at relay nodes begin to fill up, which can lead to dropped packets due to buffer overflow and/or exceeding the delay constraint. For  $\lambda > R_S$ , the network becomes overloaded, buffers overflow and packets are dropped at the source, and performance degrades more rapidly.

Our performance prediction  $S^{NC}$  provides an upper bound on the throughput of network coding and is relatively accurate for sufficiently large offered loads. For low offered loads, however,  $S^{NC}$  fails to capture the effects of the delay constraint and the inability of a relay node to collect a complete generation before it becomes blocked. Also the plots of  $S^{NC}$  and  $S^{FL}$  indicate that there is room for network coding to provide higher throughput than flooding in our scenario.

Fig. 4 clearly displays the relative performance of the Deterministic, Probabilistic, Hybrid1 and Hybrid2 rate control protocols. For  $\alpha = \beta = 1$  shown in Fig. 4(a), Probabilistic

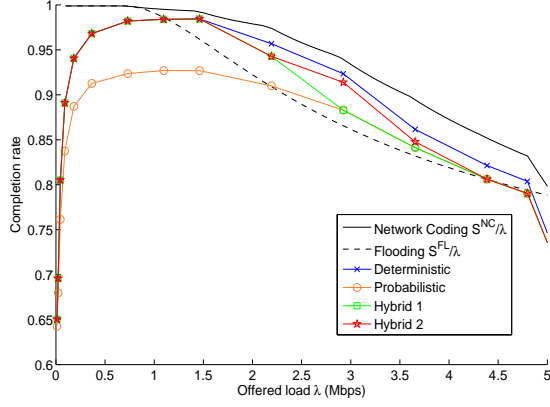
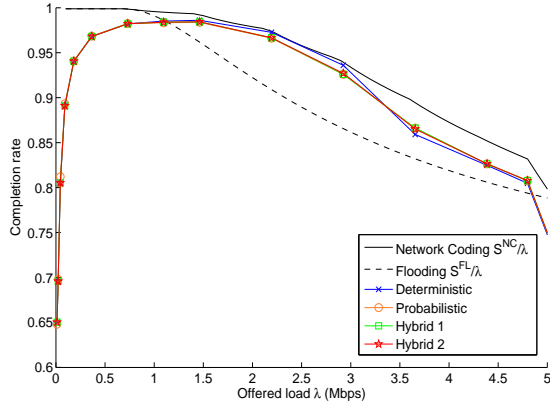
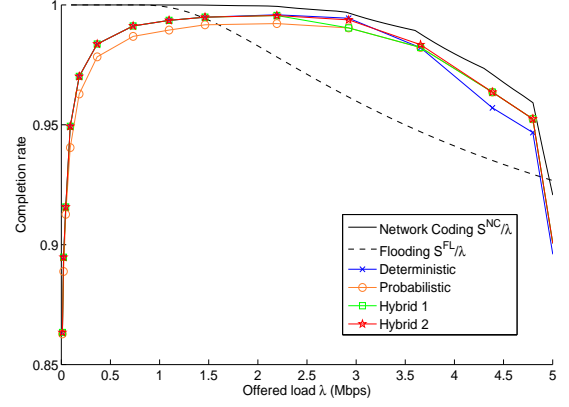
(a)  $\alpha = \beta = 1$ (b)  $\alpha = \beta = 1.2$ 

Fig. 4. Completion rate versus offered load for four protocol variations under Moderate Blockage with packet reception deadline of 2 seconds after arrival at the source.

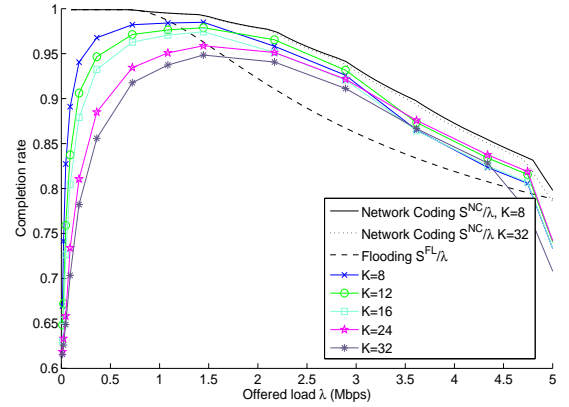
Rate Splitting suffers from a relatively large probability<sup>3</sup> that fewer than  $K$  packets are sent by relays to the destination. As expected, the Hybrid protocols provide the same performance as Deterministic when relay capacity  $C_R(t)$  is sufficient (Capacity Range 1). However, as offered load increases, the Hybrid protocols adopt Probabilistic Rate Splitting and can suffer poor performance relative to Deterministic Rate Splitting. We note here that because the Hybrid 2 Protocol utilizes more capacity than the Hybrid 1 Protocol in Capacity Range 2, it can provide improved performance. Furthermore, if the values of per-generation redundancy and rate overload factor are increased to  $\alpha = \beta = 1.2$  as shown in Fig. 4(b), the performance of Probabilistic Rate Splitting is improved.

Results in Fig. 4 indicate that the Deterministic Protocol is very effective; this can be explained by the fact that for Moderate Blockage with  $N = 6$ ,  $N(t) = 4$  is the most common case. In this case since  $K = 8$ ,  $\lceil K/N(t) \rceil = K/N(t)$  and Deterministic Rate Splitting does not unnecessarily congest the network. In contrast, Fig. 5(a) shows results for Mild Blockage

<sup>3</sup>The probability that fewer than  $K$  total packets are placed in the output queues of relay nodes is given by the probability that  $Y < K - N(t)\lfloor K/N(t) \rfloor$ , where  $Y$  is a binomially distributed random variable with parameters  $N(t)$  and  $p$ .



(a) Comparison of four protocol variations under Mild blockage with  $\alpha = \beta = 1.2$ .

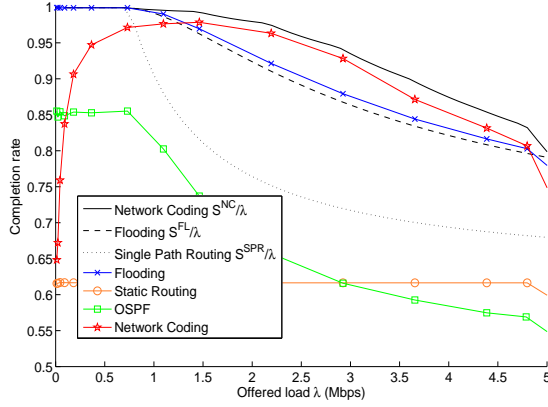


(b) Results for the Deterministic Protocol with Moderate blockage and generation sizes  $K = 8, 12, 16, 24$ , and  $32$ . Here  $B_{pkt} = 1412$  bytes and  $B_{hdr} = 52 + K$  bytes.

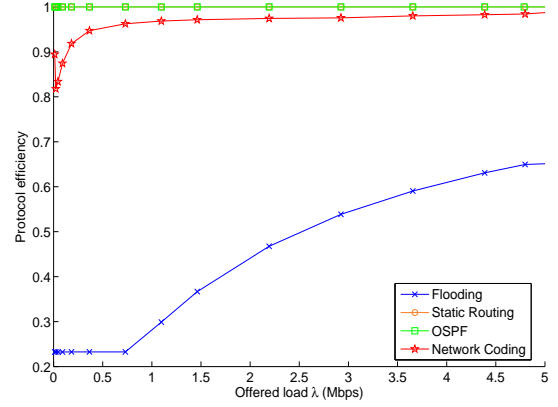
Fig. 5. Completion rate versus offered load of network coding protocols with packet reception deadline of 4 seconds after arrival at the source.

under which  $N(t) = 5$  is the most common case; in this setting Deterministic Rate Splitting dictates that  $N(t) = 5$  relay nodes will collectively send 10 coded packets for every generation of size  $K = 8$ . The resulting inefficiency of Deterministic Rate Splitting is apparent at offered loads above 3.5 Mbps.

Clear trade-offs in the choice of generation size  $K$  are shown in Fig. 5(b) for the Deterministic Protocol under Moderate Blockage. At low offered loads, larger generation sizes are less effective because of the inability of relay nodes to collect a complete generation before their link state changes and before the packet deadline expires. For this reason, up to offered loads of 1.5 Mbps, the completion rate degrades as generation size increases. Above this value of offered load, performance is determined by the difference between  $\lceil K/N(t) \rceil$  and  $K/N(t)$ . Generation sizes  $K = 12$  and  $K = 24$  provide the best performance because they are both multiples of 3 and 4, which are the most common values of  $N(t)$  under Moderate Blockage. Based on these results, we use the Deterministic protocol with  $K = 12$  for comparing network coding with other strategies under Moderate blockage.

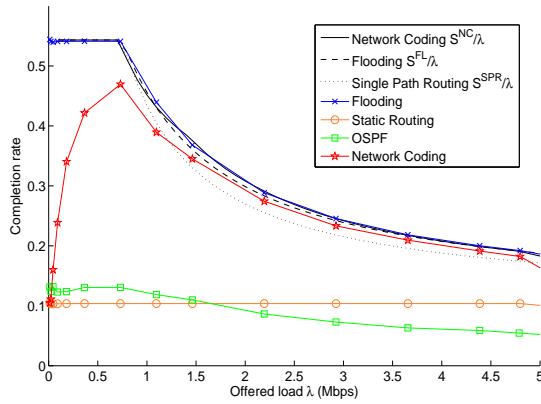


(a) Completion rate versus offered load

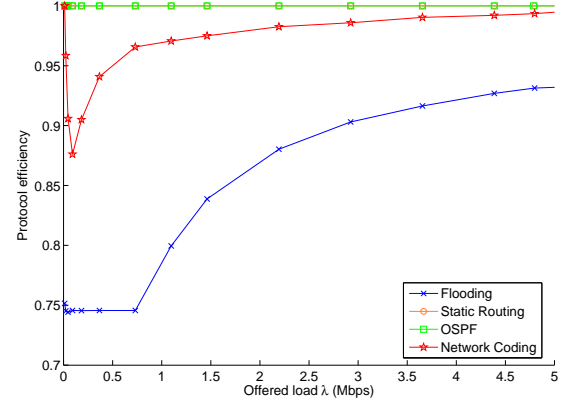


(b) Protocol efficiency versus offered load.

Fig. 6. Comparison of single path routing (Static Routing and OSPF), multipath flooding, and network coding using the Deterministic rate-control protocol with  $K = 12$  under Moderate Blockage. The packet reception deadline is 4 seconds after arrival at the source.



(a) Completion rate versus offered load.



(b) Protocol efficiency versus offered load.

Fig. 7. Comparison of single path routing (Static Routing and OSPF), multipath flooding, and network coding using the Deterministic rate-control protocol with  $K = 8$  under Severe Blockage. Relay nodes are able to collect at most 5 packets in their output buffer and the packet reception deadline is 4 seconds after arrival at the source.

### C. Comparison to single path routing and multipath flooding

Next we display results comparing network coding with other strategies, including single path routing and flooding. For single path routing, we have simulated two different strategies. First, all traffic is routed from the source node  $s$  via the satellite directly to the destination node  $d$ . In this case none of the  $N$  relay nodes ever forward traffic; we refer to this strategy as Static Routing. Additionally, we have experimented with the Open Shortest Path First (OSPF) protocol, which can adapt the route to find an unblocked path through one of the  $N$  relay nodes. To provide better functionality of OSPF in our setting, we have given strict priority to OSPF control packets and have also modified OSPF timers to allow it to more quickly identify and repair broken routes. Following the approach in [6], we have set the OSPF timers as follows: Hello Interval 1.0 sec, Dead Interval 2.0 sec, Interface Transmission Delay 0.25 sec, Retransmission Interval 2.0 sec, and SPF Calculation Delay and Hold Time to 0 sec. For flooding, each of the  $N$  relay nodes is able to receive all packets sent by the satellite node

$i$  when unblocked. The relay nodes attempt to forward all packets to the destination  $d$ ; if the output queue at a relay is full, then packets are dropped.

Results comparing single path routing, multipath flooding, and multipath network coding under Moderate Blockage are shown in Fig. 6. The completion rate results in Fig. 6(a) display the throughput performance of these different strategies. First, we note that the curve  $S^{SPR}/\lambda$  indicates that the throughput of single path routing is inherently limited in this setting due to its inability to take advantage of multiple paths. The Static Routing strategy provides constant completion rate over offered loads; the average completion rate is approximately  $p_g$ . The OSPF protocol, by adapting its path, is able to provide higher throughput than Static Routing for offered loads below 3 Mbps. However, OSPF is not able to achieve the upper bound on performance given by  $S^{SPR}/\lambda$  and actually performs worse than Static Routing for offered loads between 3 and 5 Mbps. This is due to the inability of OSPF to adapt its path quickly and accurately; there are many instances in which the destination is unblocked and a throughput of  $L_S$  bps

can be achieved, but OSPF chooses to route through a relay node and can only achieve a throughput of  $\min(L_S, R_T)$  bps. The multipath strategies outperform single-path strategies both in terms of predicted performance and implemented protocol performance. For offered loads below 1.5 Mbps, network coding suffers from incomplete generations received at relay nodes and flooding is able to provide higher throughput. However for offered loads between 1.5 and 5 Mbps, network coding is able to more efficiently utilize multiple paths. These observations are confirmed in Fig. 6(b), which plots protocol efficiency versus offered load. Since single-path protocols send each packet over at most one path, duplicate copies never propagate through the network and these strategies achieve an efficiency of one. Multipath flooding, however, provides the lowest protocol efficiency since many copies of the same packet can be sent on relay-to-destination links. Our network coding protocol achieves nearly the same efficiency as single-path routing due to the effectiveness of the rate-splitting strategy.

Results for the Severe Blockage setting are shown in Fig. 7. The predicted performance curves suggest that a genie-aided single-path strategy is able to achieve nearly the same throughput as multipath strategies. This is due to the fact that under Severe Blockage with  $N = 6$ , there is rarely more than a single unblocked path to the destination. Unfortunately, however, neither of the single-path protocols simulated are able to realize this performance. Static Routing again provides an average completion rate of approximately  $p_g$ . OSPF provides little improvement; with an average unblocked duration of 1.5 sec and a Hello Interval of 1.0 sec, the protocol is rarely able to identify an unblocked node before it becomes blocked again. Multipath flooding provides the best throughput performance among all strategies, again at the cost of lower protocol efficiency. Under Moderate Blockage we observed that network coding is able to provide higher throughput than flooding for offered loads between 1.5 and 5 Mbps, however, under Severe Blockage this is no longer true. This is also due to the fact that there is rarely more than one unblocked path to the destination; the ability of the rate splitting strategy to more effectively use paths and increase throughput does not have any impact. For this reason, between offered loads of 1.5 and 5 Mbps, network coding and flooding provide nearly the same throughput, with a slight penalty to network coding due to overhead in packet headers.

## VI. CONCLUSION

This work explores strategies to efficiently utilize bandwidth in a wireless network where the instantaneous capacity can change by orders of magnitude on time scales on the order of seconds. We focus on the use of random linear network coding to send data over multiple paths and develop rate control protocols that allow network coding to be effective in this setting. Our results indicate that when designed appropriately, network coding can provide high throughput, delay-constrained communication nearly as effectively as flooding the network while still utilizing bandwidth nearly as efficiently as single-path routing.

The results presented here point to multiple avenues for future work. First, the network topology considered here is simple in that packets traverse at most three hops from source to destination and all relay nodes are connected to the destination in one hop. Generalizing rate control to larger networks with more complicated topologies is a useful topic we plan to address in future work. Some features of our current approach, such as rate-splitting among paths by dividing up the number of packets per generation sent by relay nodes, may be easily generalized to different topologies and link rates. On the other hand, coordinating among relay nodes to determine generation IDs to be discarded for congestion avoidance may be more challenging in larger multihop networks. Also, this work proposes rate-control strategies for a single flow of traffic in the network and developing strategies to support multiple flows is necessary. For a single flow, we take the approach of estimating the (time-varying) min-cut capacity for the flow and adapting the rate to achieve it. However, it may not be possible to simultaneously achieve the min-cut capacities for multiple flows, and our approach must be adapted accordingly.

## ACKNOWLEDGMENT

The authors would like to thank Matthew Miling and Joshua Funk of OPNET Technologies for their assistance with configuration and use of the simulation software.

## REFERENCES

- [1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Proc. ACM SIGCOMM*, August 2007.
- [2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Proc. Allerton Conference*, 2003.
- [3] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 789-804, March 2006.
- [4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, August 1993.
- [5] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *Proc. IEEE International Symposium on Information Theory*, 2003.
- [6] D. Kiwior, E. G. Idhawi, and S. V. Pizzi, "Quality of service (QoS) sensitivity for the OSPF protocol in the airborne networking environment," *Proc. IEEE Military Communications Conference (MILCOM)*, 2005.
- [7] G. Lauer and D. S. Morris, "Network coding performance: an emulation experiment," *Proc. IEEE Military Communications Conference (MILCOM)*, 2008.
- [8] D. S. Lun, M. Medard, and R. Koetter, "Network coding for efficient wireless unicast," *IEEE International Zurich Seminar on Communications*, February 2006.
- [9] J.-S. Park, D. S. Lun, Y. Yi, M. Gerla, and M. Medard, "CodeCast: a network coding based ad hoc multicast protocol," *IEEE Wireless Communications Magazine*, October 2006.
- [10] B. Shrader and N. M. Jones, "Systematic wireless network coding," *IEEE Military Communications Conference (MILCOM)*, 2009.
- [11] W. M. Smith, "Channel characterization for EHF satellite communications on the move," *MIT Lincoln Laboratory Technical Report*, TR-1109, 12 July 2006.
- [12] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Network Coding for Wireless Communication Networks*, vol. 27, no. 5, pp. 622-634, June 2009.